Anomaly detection on Hypothyroidism dataset

Daniele Cecca
Matr. 918358
MSc Artificial Intelligence for Science and Technology
Email: d.cecca@campus.unimib.it

***Abstract*—This project focuses on the development and implementation of an anomaly detection system for a hypothyroidism dataset.**

Hypothyroidism is a common condition where the thyroid doesn't create and release enough thyroid hormone into your bloodstream. This makes your metabolism slow down. Also called underactive thyroid, hypothyroidism can make you feel tired, gain weight and be unable to tolerate cold temperatures.

The goal of this project is to identify strange pattern in medical data that may flag potential cases of hypothyroidism that require further investigation. The project is structures in two main parts:dataset exploration and anomaly detection

1. Introduction

Hypothyroidism happens when your thyroid doesn't create and release enough thyroid hormone into your body. This makes your metabolism slow down, affecting your entire body. Also known as underactive thyroid disease, hypothyroidism is fairly common. When your thyroid levels are extremely low, this is called myxedema. This severe type of hypothyroidism is life-threatening. In general, hypothyroidism is a very treatable condition but it can actually be difficult to diagnose it because the symptoms can be easily confused with other.

For this reason we propose an anomaly detection system that by leveraging advanced machine learning algorithms, it's able to uncover atypical patterns that may signify potential hypothyroidism cases, thus enabling earlier diagnosis and intervention.

The project is divided into two primary components: comprehensive dataset exploration and the application of multiple anomaly detection methods.

- Dataset Exploration: The first phase involves a detailed analysis of the dataset, including data preprocessing, handling missing values, and visualizing data distributions. This step is crucial for understanding the underlying characteristics of the data and preparing it for effective anomaly detection.
- Anomaly Detection: In the second phase, we apply four different anomaly detection algorithms: DBSCAN (Density-Based Spatial Clustering of Applications with Noise), LOF (Local Outlier Factor), Auto-Encoder, and MCD (Minimum Covariance Determinant).

Also we will provide a comparison and an analysis of these different methods by using different metrics and KMeans.At the end we will establish the probability of each sample of being an outlier.

2. Dataset Exploration

The dataset is on Hypothyroidism but the attributes dont' have a significant name.We can suppose that each of these variables its the result of an anamnesis.Thus they should be like symptons or features of the patient.

The dataset is composed of 7200 sample, 21 attributes each. We have 15 continuous attributes and 6 binary attributes.

We check if there were missing values but as shown in the following plot we didn't find any of them.

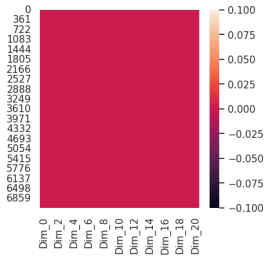


Figure 1: Heatmap missing values

Then we check for the presence of duplicates and find 71 duplicates. Even though each sample probably represents a patient, we decide to remove all duplicates and keep only the first occurrence of each sample. This is because it might be simpler for the anomaly detection algorithm to find and flag isolated points as anomalies.

2.1. Univariate analysys

We explore variables one by one. Since we have two type of attributes, we apply different methods for each type of attribute.

2.1.1. Continuous attributes. To analyze the continuous attributes we start by computing some basic statistics, summarized in the following table.

TABLE 1: Statistics of continuous variables

	Dim_0	Dim_16	Dim_17	Dim_18	Dim_19	Dim_20
count	7129.000	7129.000	7129.000	7129.000	7129.000	7129.000
mean	0.532653	0.009227	0.108475	0.179624	0.374209	0.173789
std	0.197311	0.043569	0.042208	0.060442	0.088791	0.056677
min	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000
25%	0.375000	0.001321	0.091922	0.145485	0.324074	0.143750
50%	0.562500	0.003208	0.109192	0.173913	0.365741	0.170313
75%	0.697917	0.005094	0.119777	0.205686	0.402778	0.195313
max	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000

As shown , all the continuous attributes are within the same interval [0,1] and have similar variance and mean, except the first one, Dim_0, which has values in the same interval but slightly different variance and mean. Therefore, we expect a similar distribution for all these attributes.

To have a broader and more visually inspectable view of the attributes we plot our data by using the box plot. Also the box plot will be useful to have an initial insight into the outliers of our data

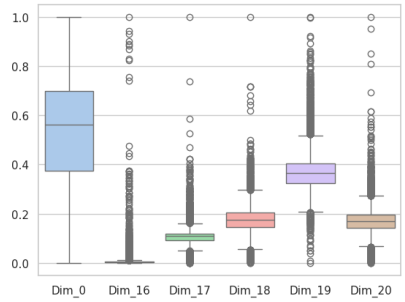


Figure 2: Box plot continuous variables

By analyzing the box plots, we can deduce that we have many possible outliers, at least from a statistical point of view. Because in general we have that outliers are identified as points with values greater than $Q3 + 1.5 \times IQR$ or less than $Q1 - 1.5 \times IQR$, where $IQR = Q3 - Q1$.

Additionally, we can already say that distributions likely have heavy tails, indicating that it probably does not follow a normal distribution. This implies high variance and the presence of heavy tails. Furthermore, the positions of the medians indicate that most distributions are not symmetric.

To validate our earlier hypothesis, we examine the distribution through histogram visualization and computation of Skewness and Kurtosis.

A near-zero Skewness indicates symmetry within the distribution. A Kurtosis exceeding 3 suggests a leptokurtic distribution, as the normal distribution maintains a value of 3.

$$\text{Skewness} = \frac{\frac{1}{n}\sum_{i=1}^{n}(x_i-\bar{x})^3}{\left(\frac{1}{n}\sum_{i=1}^{n}(x_i-\bar{x})^2\right)^{\frac{3}{2}}}$$

$$\text{Kurtosis} = \frac{\frac{1}{n}\sum_{i=1}^{n}(x_i-\bar{x})^4}{\left(\frac{1}{n}\sum_{i=1}^{n}(x_i-\bar{x})^2\right)^{2}}$$

TABLE 2: Skewness and Kurtosis of Features

Feature	Skewness	Kurtosis
Dim_0	-0.219	-0.873
Dim_16	14.454	259.349
Dim_17	3.473	45.395
Dim_18	1.510	11.000
Dim_19	1.243	4.821
Dim_20	2.373	22.437

From the results, we can conclude that they are slightly asymmetric with heavy tails, indicating that they do not follow a normal distribution.

Therefore, if we choose to use techniques such as PCA or others that assume a normal distribution of data, we should apply some transformations before using them. At lest to obtain a better result.

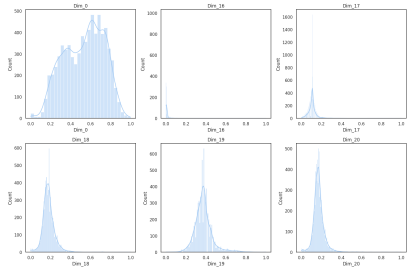


Figure 3: Histograms continuous variables

2.1.2. Binary attributes. To inspect binary features we count the number of occurrence for each variable and we plot the histogram. In this scenario, we observe a discrete distribution, specifically a Bernoulli distribution, as all variables are binary.

TABLE 3: Binary Variables

	Value 0	Value 1
Dim_1	2187	4942
Dim_2	940	6189
Dim_3	111	7018
Dim_4	91	7038
Dim_5	276	6853
Dim_6	78	7051
Dim_7	101	7028
Dim_8	121	7008
Dim_9	472	6657
Dim_10	492	6637
Dim_11	91	7038
Dim_12	59	7070
Dim_13	184	6945
Dim_14	1	7128
Dim_15	352	6777

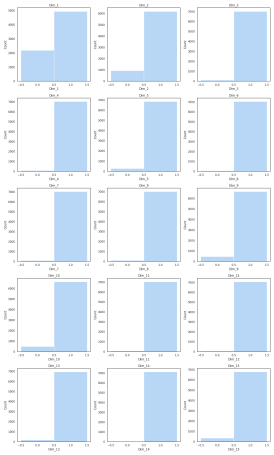


Figure 4: Histograms binary variables

As shown in the results, we have a smaller number of 0s compared to the number of 1s.

2.2. Bivariate analysis

Now, we analyze the variables pairwise to discover relationships between them. Just as we've done previously, we'll analyze continuous variables and binary variables separately.

2.2.1. Continuous variables. To capture the relationship between attributes, we could utilize correlation analysis. In this case, we decide not to standardize the variables because correlation is scale and location invariant, and also because, as we have seen before, they are within the same interval. Although one of the assumptions of the Pearson coefficient is that the variables have to be normally distributed, we can still use it because leptokurtic distributions are close to normal distributions.

$$r = \frac{\text{Cov}(X,Y)}{\sigma_X \cdot \sigma_Y}$$

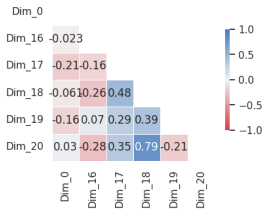


Figure 5: Correlation coefficients

We can visualize the correlations by plotting pairwise scatter plots

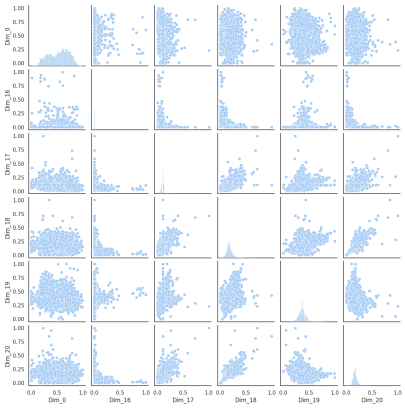


Figure 6: Pairwise scatter plots

From these results, we can deduce that all the continuous variables are important. This is because they all exhibit relatively low correlation with each other.

2.2.2. Binary variables. To compare two binary attributes, we can use the Jaccard Coefficient. If the value is equal to 1, it indicates that the two attributes provide the same information, and we could consider removing one of them.

$$J(A,B) = \frac{|A \cap B|}{|A \cup B|}$$

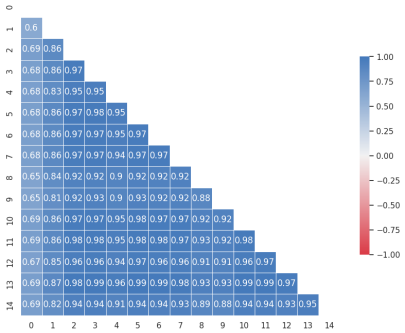


Figure 7: Jaccard coefficients

The result doesn't give a lot of information. The most informative attribute is the first one. As we already mentioned, since this concerns anomaly detection and we don't know what these variables correspond to, we prefer not to remove them, even though this analysis indicates they are identical.

2.3. Sample Analysis

Now, we analyze the relationship between samples.

2.3.1. All attributes. To determine how dissimilar two objects are, we compute a proximity matrix using Gower Distance as the distance metric.

Gower Distance is a distance measure that can be used to calculate distance between two entity whose attribute has a mixed of categorical and numerical value.

$$d_{ij} = \frac{\sum_{k=1}^{p} w_k \cdot d_k(X_{ik}, X_{jk})}{\sum_{k=1}^{p} w_k}$$

From this result, we can already see that there are some samples different from the others, as indicated by the red lines.

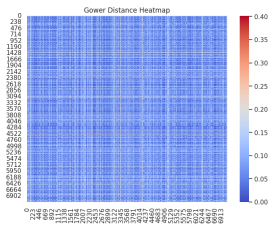


Figure 8: Proximity matrix all attributes

2.3.2. Continuous attributes. To compute the proximity matrix using only continuous attributes, we use the L1 or Manhattan distance. We opt for this distance metric because it is the same distance used in Gower for computing distances between continuous variables. It's important to note that in Gower, the distance is standardized.

$$d_{\mathrm{L1}}(\mathbf{p}, \mathbf{q}) = \sum_{i=1}^{n} |p_i - q_i|$$

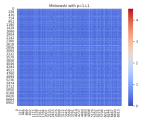


Figure 9: Proximity matrix continuous attributes

2.3.3. Binary attributes. To compute the proximity matrix using only binary attributes, we use the Hamming distance. The Hamming distance between two strings or vectors of equal length is the number of positions at which the corresponding symbols are different.

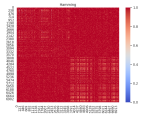


Figure 10: Proximity matrix binary attributes

From these results, we can conclude that we need both continuous and binary variables to detect outliers because both of the results doesn't give enough information.

2.4. Data visualization

To visualize the data, we cannot use PCA because our data are both continuous and binary, and PCA assumes that data are normally distributed.

Instead, we could use Factor Analysis for Mixed Data (FAMD) or t-Distributed Stochastic Neighbor Embedding (t-SNE). We decided to use t-SNE because there is no linear relationship between the variables;FAMD is better to capture these relationship. To facilitate visualization, we selected 2 components.

3. Anomaly detection

To detect the anomalies we will use four different approaches:

- DBSCAN Density-Based Spatial Clustering of Applications with Noise
- LOF ocal Outlier Fac-tor
- Auto Encoder
- MCD Minimum Covariance Determinant

For most of these algorithms, we need to specify the fraction of outliers (contamination). Therefore, as the first method, we choose DBSCAN, which does not require this parameter.

3.1. DBSCAN

DBSCAN - Density-Based Spatial Clustering of Applications with Noise. Finds core samples of high density and expands clusters from them.

DBSCAN depends on two parameters:

- MinPts: Minimum number of points required to form a dense region (core point).
- Eps: Maximum distance that specifies the neighborhood of a point, which determines its core points and directly density-reachable points

To determine Eps, we compute the distance from each point to its 4th nearest neighbor using k-nearest neighbors (KNN), and then sort these distances.

As the distance metric, we will use the Gower distance. Therefore, we will reuse the proximity matrix computed previously.

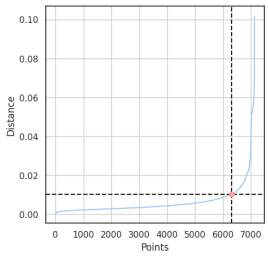


Figure 11: Sorted distanced 4th

As we can see from the previous image we select as eps 0.01. Then we apply the model and we have obtained these results:

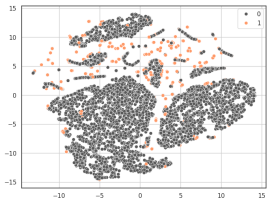


Figure 12: Output DBSCAN

- Number of normal data: 6422
- Number of outlier data: 707

Because DBSCAN found 620 outliers we will use a contamination of 10% for the next models.

3.2. LOF

The anomaly score of each sample is called Local Outlier Factor. It measures the local deviation of density of a given sample with respect to its neighbors. It is local in that the anomaly score depends on how isolated the object is with respect to the surrounding neighborhood. More precisely, locality is given by k-nearest neighbors, whose distance is used to estimate the local density. By comparing the local density of a sample to the local densities of its neighbors, one can identify samples that have a substantially lower density than their neighbors. These are considered outliers.

To choose the number of neighbors, we use the thumb rule, which suggests taking 20% of the data. As the distance metric, we will use again Gower.

We obtain the following results:

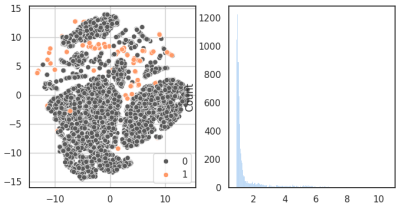


Figure 13: Output LOF

- Number of normal data: 6489
- Number of outlier data: 640

We didn't plot only data but also the histogram of anomaly scores.This is useful for observing if the different methods exhibit similar behavior and if in general they behave in a reasonable manner.

3.3. Auto Encoder

Auto Encoder (AE) is a type of neural networks for learning useful data representations unsupervisedly. Similar to PCA, AE could be used to detect outlying objects in the data by calculating the reconstruction errors.

TABLE 4: Neural Network Architecture Summary

Layer (type)	Output Shape	Param #
Dense (Dense)	(None, 21)	462
Dropout (Dropout)	(None, 21)	0
Dense_1 (Dense)	(None, 21)	462
Dropout_1 (Dropout)	(None, 21)	0
Dense_2 (Dense)	(None, 18)	396
Dropout_2 (Dropout)	(None, 18)	0
Dense_3 (Dense)	(None, 9)	171
Dropout_3 (Dropout)	(None, 9)	0
Dense_4 (Dense)	(None, 9)	90
Dropout_4 (Dropout)	(None, 9)	0
Dense_5 (Dense)	(None, 18)	180
Dropout_5 (Dropout)	(None, 18)	0
Dense_6 (Dense)	(None, 21)	399

By applying it we obtain the following results:

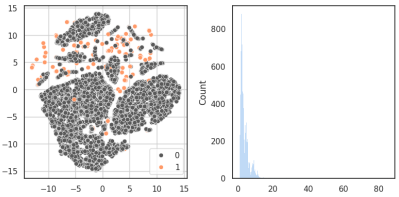


Figure 14: Output Auto-Encoder

- Numeber of normal data: 6416
- Number of outlier data: 713

3.4. MCD

Even though we have observed that analyzing samples with only continuous attributes does not reveal any distinctive differences between them, we have decided to use the Minimum Covariance Determinant method. This is because it is a statistical method and could provide a different perspective on outliers.

The main idea of the MCD is that for a fixed h number of samples (support_fraction), with $\frac{n\_sample+n\_features+1}{2} \leq h \leq n$, in our case $\frac{n\_sample+n\_features+1}{2}$, it computes:

- $\hat{\mu}$: the mean of the h observations for which the determinant of the sample covariance matrix is minimal;
- $\hat{\Sigma}$: the minimal covariance matrix.

Then compute the Mahalanobis distance as the outlier degree of the data.

By applying it we obtain the following results. As seen from the plot, the points flagged as outliers by MCD are completely different from those identified by the other methods. Therefore, we will not use this method in the next analysis.

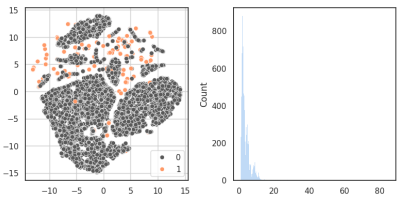


Figure 15: Output Auto-Encoder

- Number of normal data: 6416
- Number of outlier data: 713

4. Check coherence between outlier detection models

To check the difference between the models we compare the histograms by using the mutual information and we compare the labels by using Jaccard score.

The mutual information score indicates the amount of information shared between the two histograms: a higher score suggests more shared information, while a lower score suggests less shared information.

$$I(X;Y) = H(X) + H(Y) - H(X,Y)$$

To compare the scores, since they represent two different metrics, namely the local outlier factor and reconstruction error, we standardize the values by using Min-MaxS scaler.So they fall within the interval [0,1]

$$x' = \frac{x - x_{\min}}{x_{\max} - x_{\min}}$$

As shown in the results, each model identifies quite different outliers. However, LOF and the autoencoder are the two methods that have the most labels in common."

- Mutual Information Score: 0.272
- Jaccard Coefficient LOF-Encoder 0.56
- Jaccard Coefficient LOF-DBSCAN 0.22
- Jaccard Coefficient Encoder-DBSCAN 0.36

5. Delete Outlier

After the analysis of the different methods we remove the outliers from the dataset. and we create four different dataset:

1) df_lof: dataframe without the outliers found by LOF
2) df_encoder: dataframe without the outliers found by Auto-Encoder
3) df_dbscan: dataframe without the outliers found by DBSCAN
4) df_encoder_lof: dataframe without the outliers found in common between LOF and Auto-Encoder.

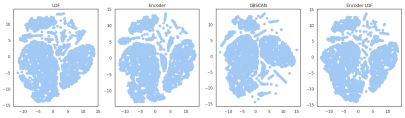


Figure 16: Dataframes without outliers

6. Clustering

To evaluate the different methods of outlier detection, we apply k-means to the different dataframes and assess the results using the Silhouette Score.

$$silhouette(i) = \frac{b(i) - a(i)}{\max(a(i), b(i))}$$

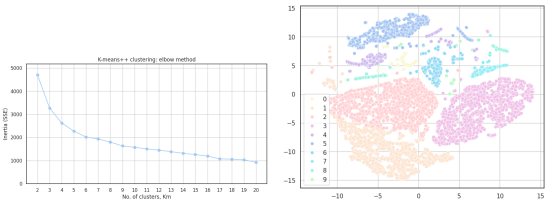
Although our dataset may not be well-suited for k-means due to its small variance, we use this method only as a reference point. We expect a poor result across all methods, but we can still determine the best one.

Note that if we want to determine the "correct" number of clusters we should explore also the results by using metrics like BIC AIC or adjust R square that consider the number of clusters in their computation.

6.1. K-Means

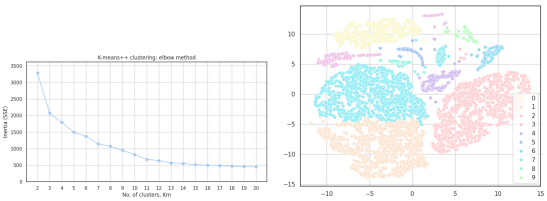
To determine the number of cluster we use the elbow method. We use K-means ++ because it suffers less from the initialization problem. Although the number of cluster is not really important in this case, and we always use the same number of cluster (k=10), for the reason explained previously.

6.1.1. K-Means all data. .



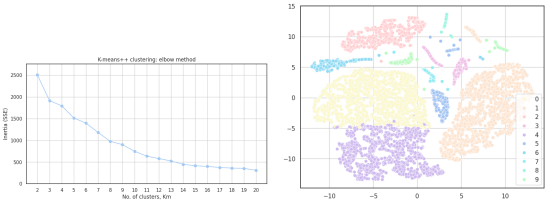
(a) Elbow all data (b) K-means all data

6.1.2. K-Means LOF. .



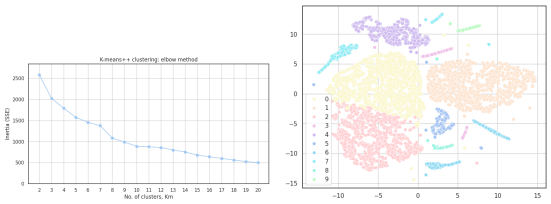
(a) Elbow lof data (b) K-means lof data

6.1.3. K-Means Auto-Encoder. .



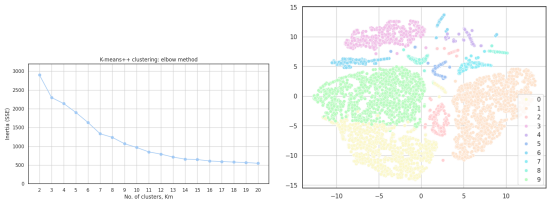
(a) Elbow encoder data (b) K-means encoder data

6.1.4. K-Means DBSCAN. .



(a) Elbow dbscan data (b) K-means dbscan data

6.1.5. K-Means LOF-Encoder. .



(a) Elbow lof-encoder (b) K-means lof-encoder

7. Comparison

As anticipated, the results are not very satisfactory. However, in all cases, we observe improved performance compared to the dataset including all data points.

The LOF algorithm demonstrates the best performance.

It's worth noting that the dataframe excluding outliers identified by both LOF and auto-encoder also performs well compared to the others, even though it removes fewer outliers.

TABLE 5: Silhouette Scores per Model

Model	Silhouette Score
Kmeans-LOF	0.458
Kmeans-Encoder	0.453
Kmeans-DBSCAN	0.451
Kmeans-LOF Encoder	0.444
Kmeans	0.387

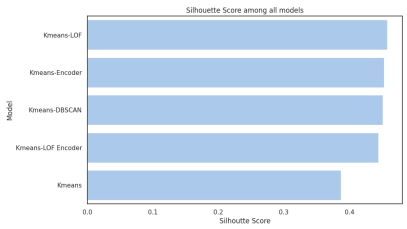


Figure 22: Comparison

8. Probability computation

At the end, we decide to assign to each sample the probability of the data object being anomalous. We compute the following probability: $P(A = \text{anomalous} \mid X_1, X_2, \ldots, X_n)$

To compute this score, we will use Bayes' Theorem: $P(A = \text{anomalous} \mid X) = \frac{P(X \mid A=\text{anomalous}) \cdot P(A=\text{anomalous})}{P(X)}$

where

- $X = \{X_1, X_2, \ldots, X_n\}$.
- $P(X_1, X_2, \ldots, X_n) = P(X \mid A = \text{anomalous}) \cdot P(A = \text{anomalous}) + P(X \mid A = \text{normal}) \cdot P(A = \text{normal})$

Computing the prior probability $P(A = \text{anomalous})$ is straightforward; it's the number of outliers divided by the total number of samples.

To compute $P(X_1, X_2, \ldots, X_n \mid A)$, we use anomaly scores computed by LOF (Local Outlier Factor). We chose LOF because it performed best among all models we analyzed.

To transform our score into a probability, there are three potential methods:

Computing the prior probability $P(A = \text{anomalous})$ is straightforward; it's the number of outliers divided by the total number of samples.

To compute $P(X_1, X_2, \ldots, X_n \mid A)$, we use anomaly scores computed by LOF (Local Outlier Factor). We chose LOF because it performed best among all models we analyzed.

To transform our score into a probability, there are three potential methods:

1) We could model our score to a distribution, if we know the distribution of the scores.
2) We could use Kernel Density Estimation.
3) We can use the sigmoid function.

We opted for the sigmoid function because it is simpler and does not require knowledge of the score distribution. Therefore, we do not need to compute Bayes' Theorem directly and can apply the sigmoid function directly. $\sigma(x) = \frac{1}{1+e^{-x}}$

When we use the sigmoid function, it is crucial to standardize our values to prevent the function from compressing the values too much, especially if the scores have high variance. $z = \frac{x-\mu}{\sigma}$

After the standardization and the application of the sigmoid we obtain the following probabilities:

TABLE 6: Probability 10 sample

Sample ID	Anomaly	Anomaly Probability
3012	0	0.368795
6225	0	0.369883
5399	0	0.369970
4255	0	0.371542
405	1	0.903388
6999	0	0.515087
3419	0	0.379816
3715	0	0.374490
4803	0	0.677050
2472	0	0.379833

9. Conclusion

In this work, we conducted a comprehensive analysis of the dataset. We began with a univariate analysis of the attributes, followed by a bivariate analysis, and concluded with a more detailed microscopic examination by analyzing the relationships between samples.

To perform each of these stages, we used different techniques based on the type of attributes and the characteristics of the dataset. After this analysis, we applied various anomaly detection models and compared their results using different techniques such as the Jaccard Score and Mutual Information. Furthermore, we compared the results by applying k-means and inspecting the outcome using the Silhouette Score.

Once we established the best model, we computed the probability of a given sample being an outlier.

Overall, we proposed a broad application of anomaly detection techniques. However, to better explain and understand the results, it would be beneficial to have more knowledge about the attributes and a deeper understanding of the domain.

Since outlier detection often suffers from model instability due to its unsupervised nature, future work could involve exploring other models or trying ensemble methods. Additionally, we could attempt to build a Bayesian network using the probabilities computed from the anomaly scores.

References

[1] Cleveland Clinic, *Hypothyroidism*, https://my.clevelandclinic.org/health/diseases/12120-hypothyroidism, accessed June 13, 2024.

[2] scikit-learn, *Nearest Neighbors*, https://scikit-learn.org/stable/modules/neighbors.html

[3] Scikit-learn Developers, *DBSCAN*, https://scikit-learn.org/stable/modules/generated/sklearn.cluster.DBSCAN.html

[4] PyOD Developers, *LOF*, https://pyod.readthedocs.io/en/latest/pyod.models.html#module-pyod.models.lof

[5] PyOD Developers, *Auto Encoder*, https://pyod.readthedocs.io/en/latest/pyod.models.html#module-pyod.models.auto_encoder

[6] PyOD Developers, *Minimum Covariance Determinant (MCD)*, https://pyod.readthedocs.io/en/latest/pyod.models.html#module-pyod.models.mcd

[7] Scikit-learn Developers, *sklearn.metrics.silhouette_score*, https://scikit-learn.org/stable/modules/generated/sklearn.metrics.silhouette_score.html

[8] scikit-learn, *sklearn.metrics.mutual_info_score*, https://scikit-learn.org/stable/modules/generated/sklearn.metrics.mutual_info_score.html

[9] Scikit-learn Developers, *sklearn.metrics.jaccard_score*, https://scikit-learn.org/stable/modules/generated/sklearn.metrics.jaccard_score.html

[10] Python Package Index (PyPI), *Gower*, https://pypi.org/project/gower/

[11] scikit-learn, *sklearn.cluster.KMeans*, https://scikit-learn.org/stable/modules/generated/sklearn.cluster.KMeans.html

[12] scikit-learn, *sklearn.preprocessing.StandardScaler*, https://scikit-learn.org/stable/modules/generated/sklearn.preprocessing.StandardScaler.html

[13] scikit-learn, *sklearn.preprocessing.MinMaxScaler*, https://scikit-learn.org/stable/modules/generated/sklearn.preprocessing.MinMaxScaler.html

[14] Towards Data Science, *Why Sigmoid? A Probabilistic Perspective*, https://towardsdatascience.com/why-sigmoid-a-probabilistic-perspective-42751d82686

[15] Notes of the course of advanced statitistic, *Box-Plot,Minimum Covariance Determinant (MCD),standardize methods, Distributions*,

[16] Notes of the course of unsupervised learning, ,

10. Disclosure Statement

The authors declare that this report is entirely original and does not contain any plagiarism.